



# Agile EVM

22<sup>nd</sup> Annual Systems & Software Technology Conference  
(SSTC)  
Salt Lake City, Utah  
April 2010

Mr. J. Matthew Park  
Northrop Grumman Corporation  
matt.park@ngc.com

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>APR 2010</b>		2. REPORT TYPE		3. DATES COVERED <b>00-00-2010 to 00-00-2010</b>	
4. TITLE AND SUBTITLE <b>Agile EVM</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>Northrop Grumman, 2980 Fairview Park Drive, Falls Church, VA, 22042</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES <b>Presented at the 22nd Systems and Software Technology Conference (SSTC), 26-29 April 2010, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License</b>					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>Same as Report (SAR)</b>	18. NUMBER OF PAGES <b>17</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

# Audience Assumptions For This Presentation

---



- Software projects exist for which agile development process provide a greater chance of success than a traditional waterfall life-cycle
- Agile is not an excuse for ignoring processes, management, or engineering
- Agile produces useful programmatic metrics

# Four Questions From Implementing Agile EVM

---



- “What is different about Agile EVM from EVM”
- “Agile metrics and charts provide insight into progress, but how is the project proceeding according to the plan?”
- “How do I report SPI and CPI for an agile project?”
- “How do I get credit for work partially completed in a fixed timebox?”



# Agile Requirements

## Traditional Systems Engineering



Many Layers of Functionally Decomposed  
Requirements Ending in Atomically  
Described Functions

## Agile Systems Engineering

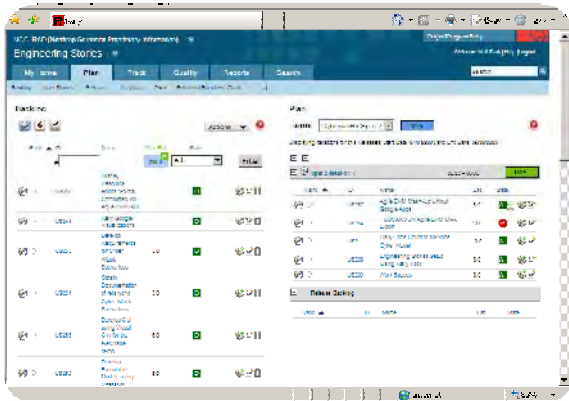


Many Layers of Functionally Decomposed  
Requirements Ending in Deliverable  
Capabilities Useful to User



# Key Differences – Part 1

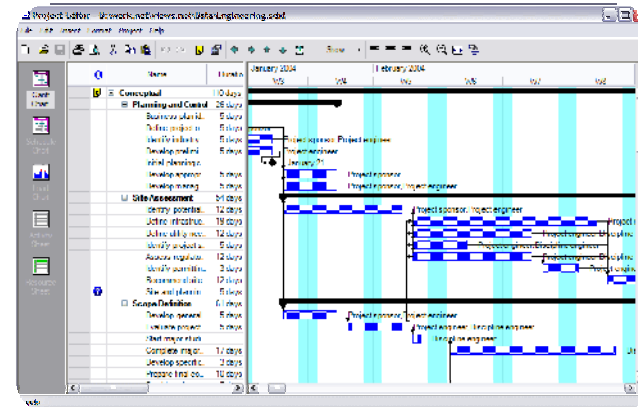
## Agile



### Focused on Delivering Capabilities

- Manage based on delivery of capabilities scheduled into fixed time-boxes
- Identify tasks required to deliver each story
- Manage risks while identifying all tasks necessary to complete capabilities

## EVM



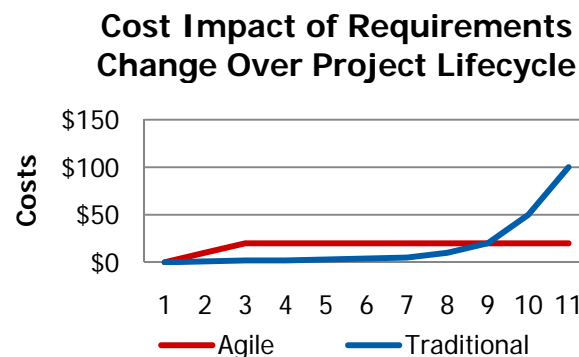
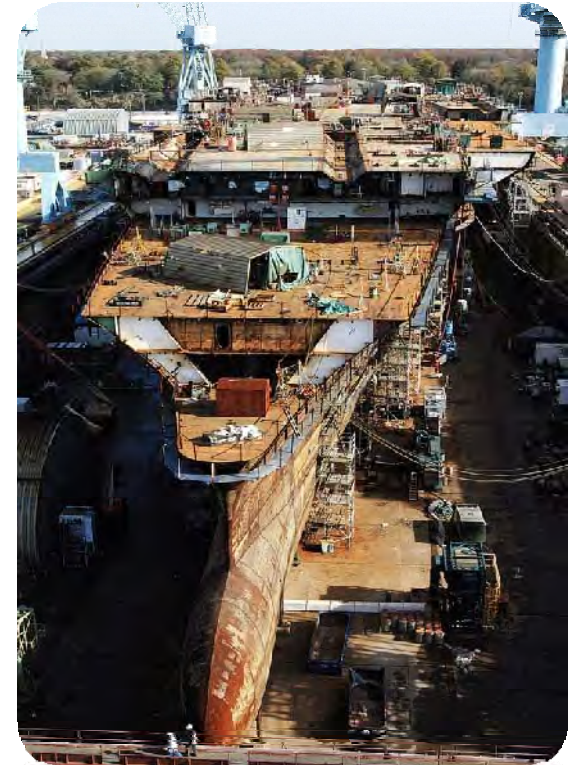
### Focused on Completing Activities

- Manage based on the completion of activities broken down into small duration tasks
- Manage risks while identifying all tasks necessary to complete the system requirements



# Key Differences – Part 2

- Agile planning balances the effort and investment in planning with the knowledge that the plan will be revised through the course of the project
  - Well suited for 6 to 18 month long projects or increments of larger programs
  - Allows requirements to flex to meet business and mission needs
  - Typically does not perform complete planning, scheduling, and costing of a project to discrete work packages to the same level of details as traditional EVM efforts



- What does it cost to change the width of an aircraft carrier?
- How would ship design change if it was inexpensive?

# Key Differences – Part 2 – Balances Constraints





# Key Differences – Part 3 & 4

---

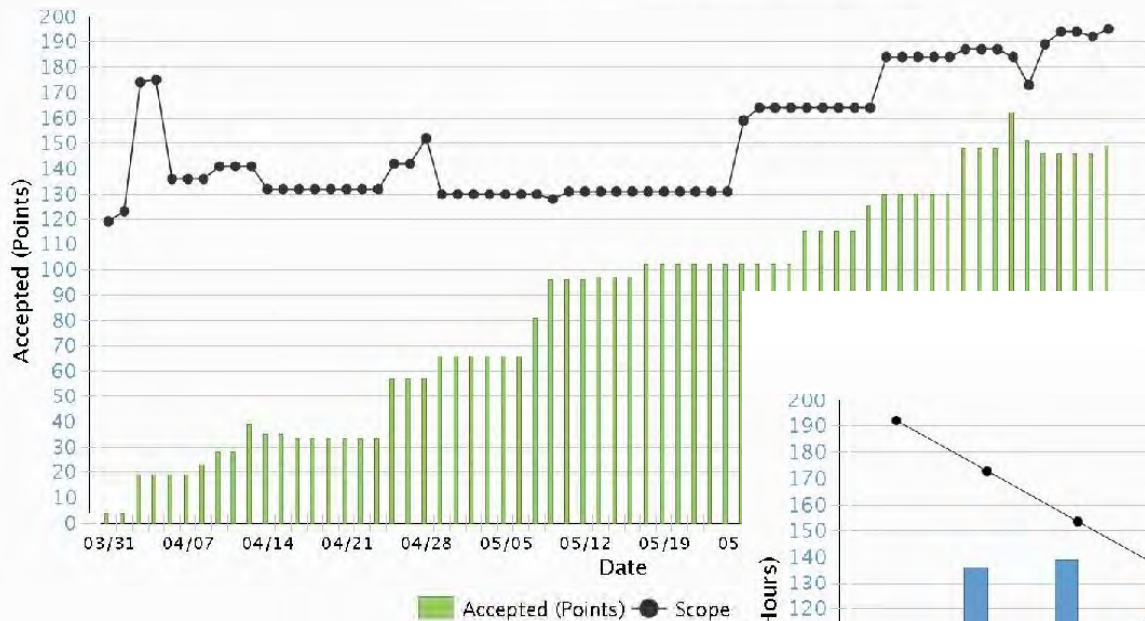


- Project teams commit to delivering specified amount of capabilities or requirements in a fixed time versus tasks in a specified time
  - Measured in story points
  - Establish “Velocity” of story points per time (i.e. iteration or release)
- Agile takes credit for integrated, and delivered code while EVM credits value when tasks are complete
  - Agile uses a customer centric definition of value aligned to features and requirements over programmatic activities
  - “Value delivered earlier”

# Agile Progress Charts – “Something is “Missing

Release Burnup

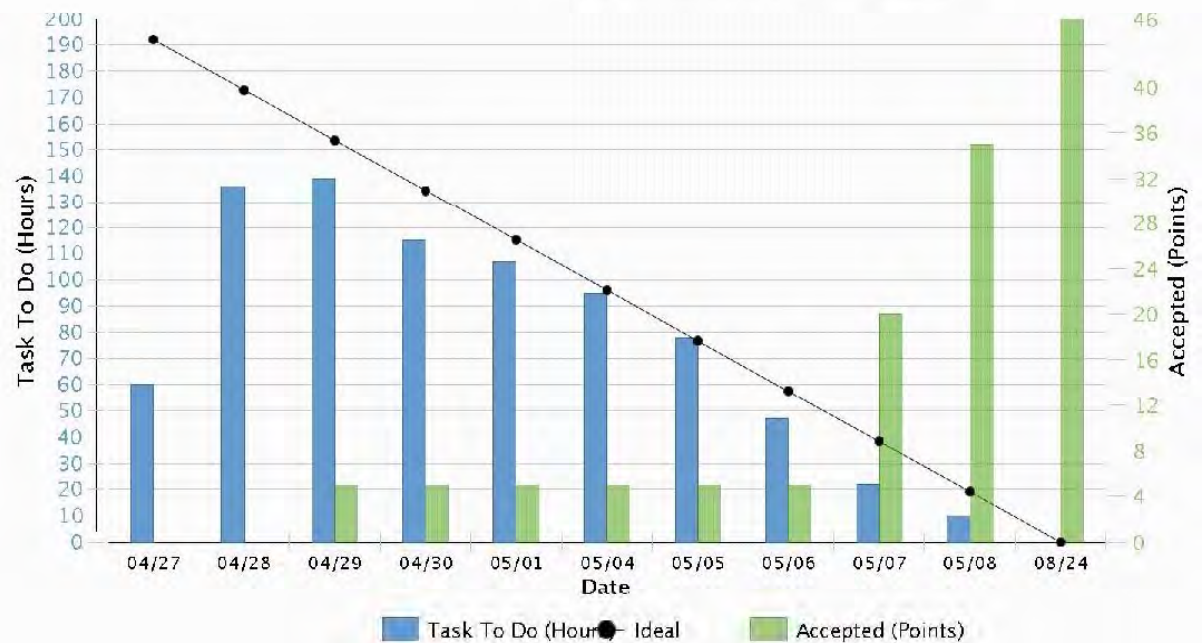
Spin 2 (04/01/09 – 06/30/09)



“That is great, but what was the original plan?”

Iteration Burndown

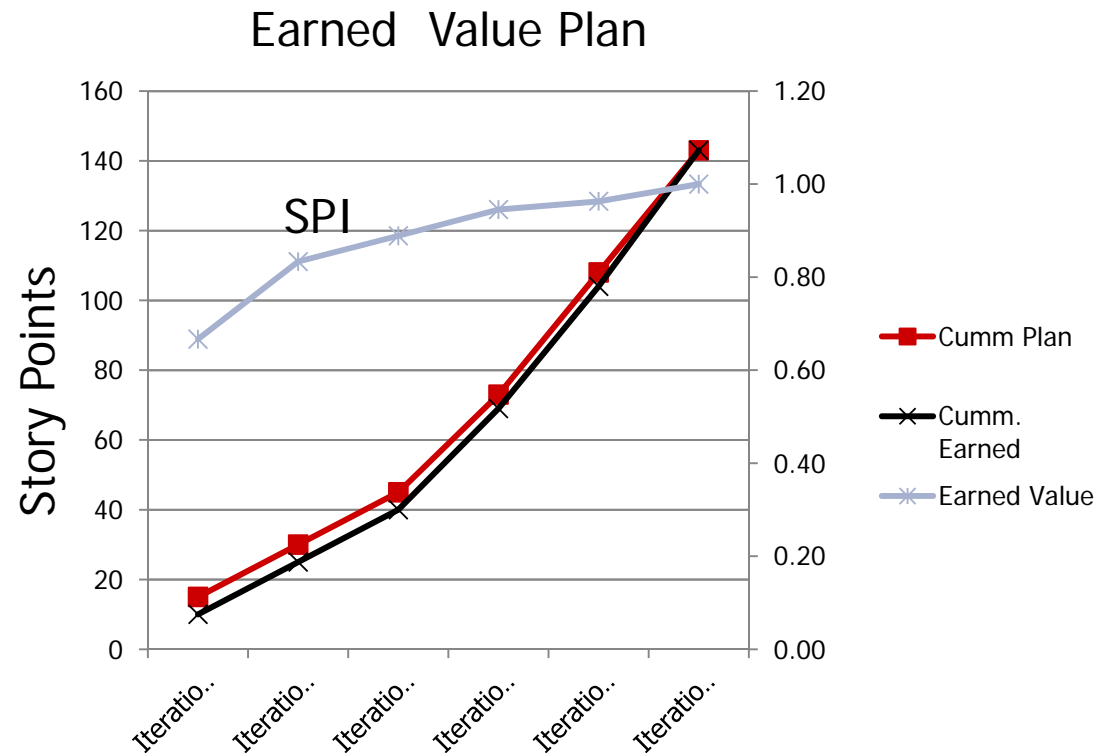
Spin 2 Iteration 3 (04/27/09 – 05/08/09)



= Recommendations

# Planning Scope For Agile – “Filling EVM Gap”

- Plan for 6 to 18 month durations
- Baseline project capacity in story points
  - Project level (12 months)
  - Release/spin level (3 months)
  - Iteration level (2 to 4 weeks)
- Track and calculate progress (SPI) against the story point plan
- Get Engineering Team participation



# New Way Of Approaching SPI

- How to get less than or greater to 1.0 in a fixed duration timebox ?
  - Traditionally
    - “Complete scheduled tasks slower or faster than planned.”
  - Agile EVM
    - “Deliver or complete more or less work than planned.”
- Why
  - Agile uses fixed timeboxes to schedule work and varies the amount of scope in a timebox





# When to Replan and When to Take Credit for 1.0+ SPI?

- Scenario
  - A project is able to take on more work than originally planned in a release
- Agile EVM Recommendation
  - Move stories forward or add stories from existing project backlog (SPI > 1.0)
  - Replan to increased scope if added stories are not in current backlog ( SPI = 1.0)





- Allow late acceptance for small amounts of remaining work (< 10% iteration)
- Split story points on release boundaries for large amounts of carrying over work for accepting stories
- SPI is measured for the entire "contractual" effort (i.e. Increment X)
- Deferred or antiquated requirements are removed from backlog thus reducing the total story points or replaced with new requirement of equal story point value

# When to Take Credit For Value

- When
  - Development complete?
  - Working in field?
  - Integration and testing complete?
  - After certification and accreditation?
- Many domains have extensive external certification processes that delay actual fielding
- Recommendation
  - If delivery and deployment time is significantly less than the development and testing time
    - Take credit for value after demonstration in the field
  - Else
    - Take credit after an integration and test milestone that demonstrates acceptable levels of confidence
    - Defect process will account for rework
    - Iterative aspect of agile reduces risk of costly defects leaking through to field

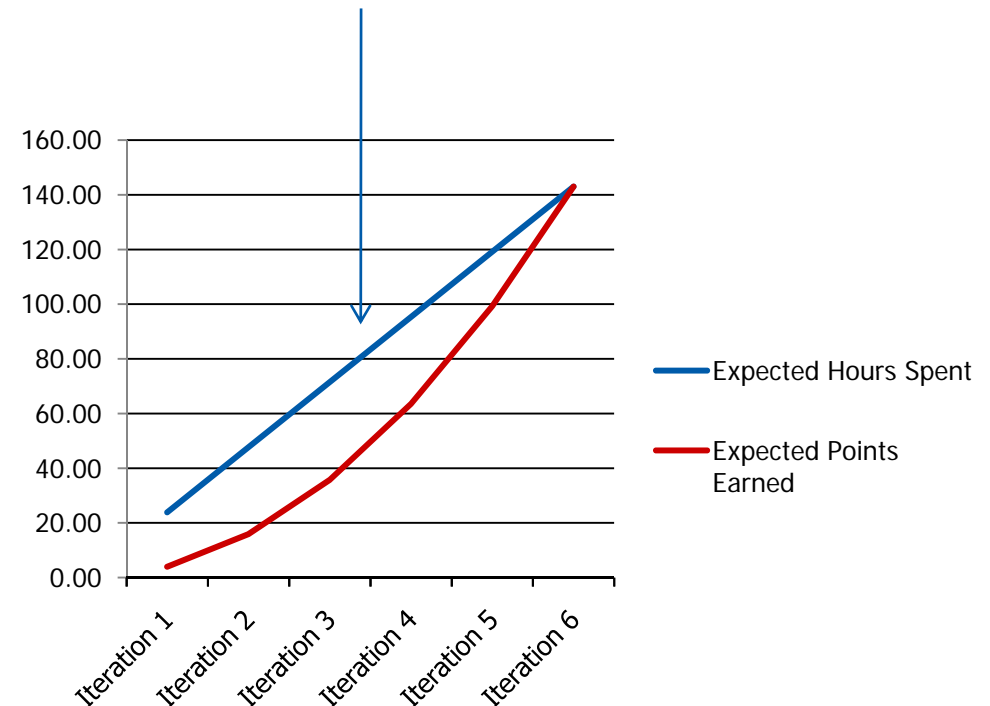


# When to Take Credit For Value – Partial Credit

- “Pure” Agile does not give partial credit
  - Similar to EVM’s 0,100 credit concept
- 0,100 Model can cause undesirable variances for projects where delivery time is large with respect to development time
  - Good candidate for 0,50,100 or 0,85,100 approach for credit



Try to avoid a reporting process that likely cause variances







- Agile EVM Papers, experience, and tools support that Agile can provide metrics that feed a more traditional EVM approach
  - “AgileEVM – Earned Value Management in Scrum Projects”, Suliarman, Barton, and Blackburn
- Agile EVM takes a mind-set adjustment focused on planning, delivering, and reporting on value earned for capabilities
- Agile EVM is a minimal project burden for disciplined agile projects

***NORTHROP GRUMMAN***



DEFINING THE FUTURE